# CMPE 124 Lab 8: Flip flop and latch circuits

Leonardo Blas

***Abstract*—The purpose of this lab project is to implement several latch and a flip flop digital circuits via logic gates or existing parts (such as a 74_74 D-type flip flop).**

## I. Introduction

This lab project is based on the implementation of SR latch circuits using NOR and NAND gates, as well as a controlled SR latch and D latch, and an edge triggered D flipflop using a 74_74 and flipflops circuits. Furthermore, note that while latches don't require of clock and clear inputs, flip flops do.

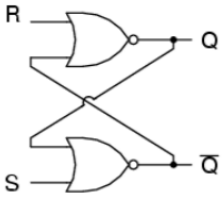
## II. Design Methodology

For this lab project, we implement 4 latch digital circuits: SR latch circuits using NOR and NAND gates, and a controlled SR latch and a D latch circuits. Additionally, we also implement a flip flop circuit: an edge triggered D flipflop using a 74_74 and flipflops circuits.
Each of these circuits depends on different input, and while some have different number of inputs than others, in the end they all result in the same output, q and not q. For further implementation details, see the latch circuits' diagrams and their truth tables in the upcoming figures.

### *A. Parts List*

- Clock
- 5V source
- 74_74
- Binary switches
- Resistor
- NAND gates
- NOR gates
- NOT gates

### *B. Truth Tables*




| S | R | Q | $\overline{Q}$ |
|---|---|---|---|
| 0 | 0 | latch | latch |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

**Figure 1. Truth table for a SR latch using NOR gates circuit.**

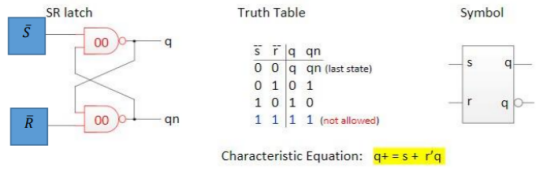
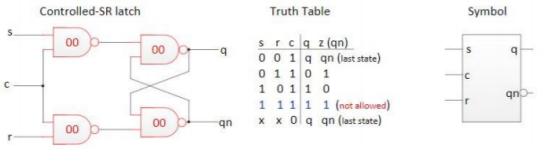



**Figure 2. Truth table for a SR latch using NAND gates circuit.**



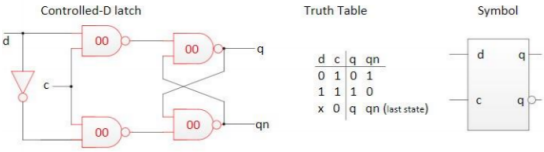




**Figure 3. Truth table for the controlled SR and D latch circuits.**

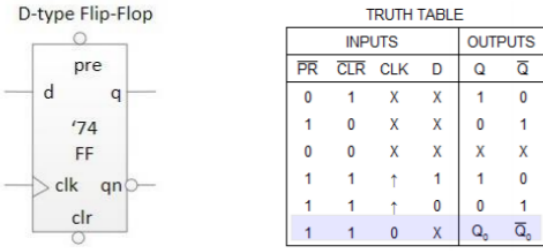



**Figure 4. Truth table for an edge triggered D flipflop using a 74_74 and flipflops.**

### *C. Schematics*

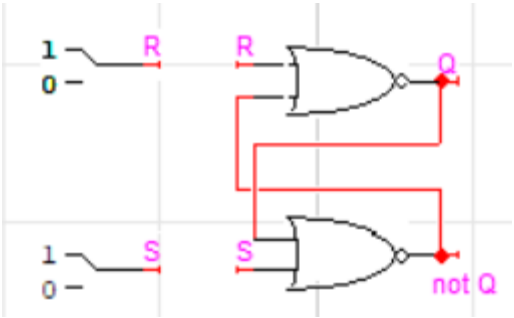



**Figure 5. SR latch using NOR gates.**

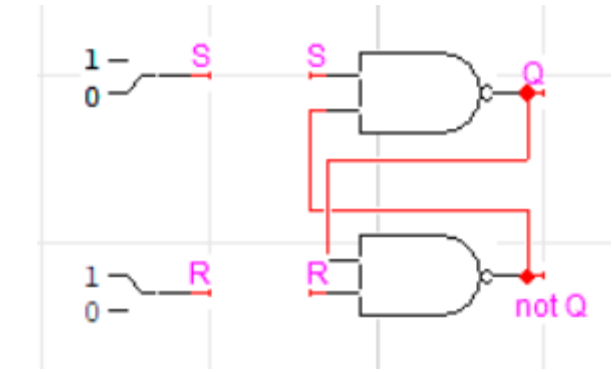



**Figure 6. SR latch using NAND gates.**

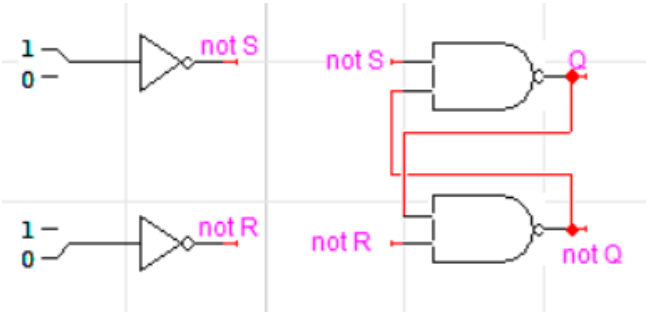



**Figure 7. Controlled-SR latch.**

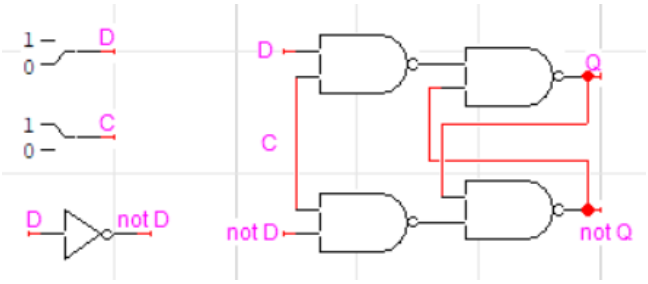



**Figure 8. Controlled-D latch.**

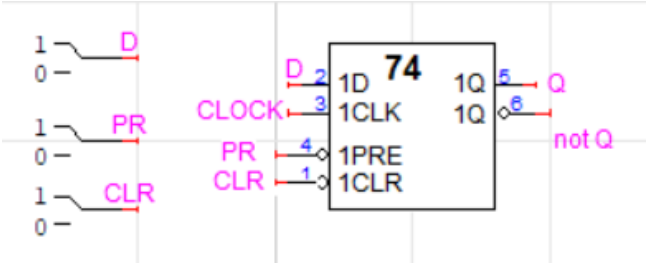



**Figure 9. Edge triggered D flipflop using a 74_74 and flipflops.**

## III. TESTING PROCEDURES

1. Create the five circuits, as portrayed in Figures 1 through 9.
2. Set up the binary switches to test each circuit's truth tables.
3. Reset the binary switches and run the simulation.
4. Record each circuits' waveform.

## IV. TESTING RESULTS

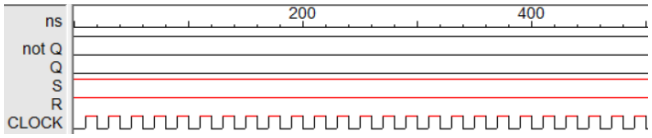



**Figure 10. SR latch using NOR gates' sample waveform.**

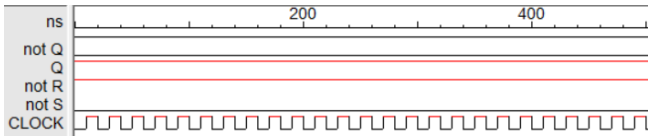



**Figure 11. SR latch using NAND gates' sample waveform.**

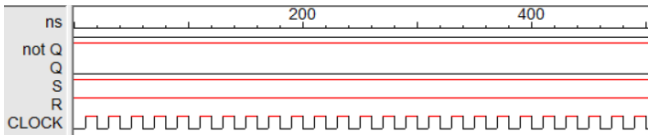



**Figure 12. Controlled-SR latch's sample waveform.**

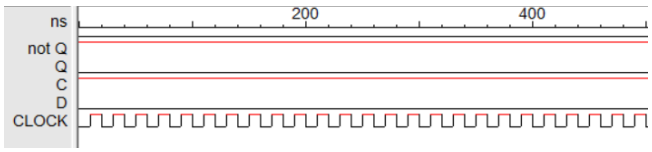



**Figure 13. Controlled-D latch's sample waveform.**

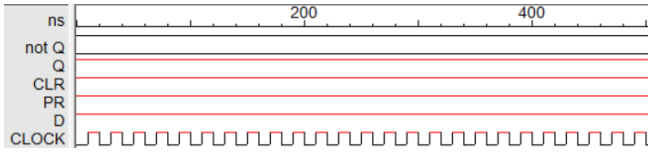



**Figure 14. Edge triggered D flipflop using a 74_74 and flipflops' sample waveform.**

## V. CONCLUSION

In conclusion, there are several types of latch and flip flop circuits that we can implement and experiment with, in order to further understand how a computer's memory works. While apparently simple circuits, latches can prove to be very powerful, as they are able to store binary data and many latches can be put together to serve as a computer's physical memory. Flip flops circuits, while very similar to latch circuits and also able to store binary data, differ by requiring clear and clock inputs.
Furthermore, after checking our sample waveforms against their corresponding truth tables, we learned that we successfully implemented our latch and flip flops circuit. Finally, this lab has taught us the basics behind the implementation of a computer's physical memory, which not only tests our theoretical knowledge—obtained in our lectures—on binary data, digital signals, and logic gates, but also advances our comprehension of logic circuits and how computers operate.